Stephen Waldron

COP 3538 Data Structures using OOP

Dr. Roggio

Project 5 Pseudocode HashTable class

**HashTable class**

**public HashTable ()**

for loop the count through the elements of hashTable3 array

Declares a new LinkedList object and assigns it to hashTable3 at index value of count

end HashTable constructor

**public void buildLetterAndValueArrays (String input)**

declare inputSplit String array and assign input string calling split method and passing a string of a comma to the method

assign inputSplit array at index 0 to letterArray at letterArrayCounter and increment by 1

assign inputSplit array at index 1 the parsed integer value valueArray at valueArrayCounter and increment by 1

end buildLetterAndValueArrays method

**public int getHashValueForTable1 (String name)**

declare hashValue variable of type int and assign the value of 0 to the variable

declare nameSplit String array and assign the input string calling the split method and passing an empty string to the method

declare character variable of type String

for loop to count through the indexes of nameSplit array

assign nameSplit at index value of count to character

for loop to count through the indexes of the letterArray array

if character equals ignore case letterArray at index count1

add the current value of hashValue to the integer value of valueArray at index count1 and assign to hashValue

end for loop for letterArray

end for loop for nameSplit

return the value of hashValue mod by the value of hashTable1 length to the calling environment

end getHashValueForTable1 method

**public void buildHashTable1 (Presidents thePresident)**

declare a new TableDataItems object called tableData and pass thePresident to the constructor

declare hashValue variable and assign the return from getHashValueForTable1

while hashTable1 at hashValue not equal null and hashTable1 at hashValue not logically deleted

add 1 to the value of hashValue and assign to hashValue

if hashValue equals to hashTable1 length

mod hashValue by hashTable1 length (to do a wraparound) and assign to hashValue

end while loop

assign tableData to hashTable1 at hashValue index

end buildHashTable1 method

**public int getHashValueForTable2 (String name)**

declare hashValue variable of type int and assign the value of 0 to the variable

declare nameSplit String array and assign the input string calling the split method and passing an empty string to the method

declare character variable of type String

for loop to count through the indexes of nameSplit array

assign nameSplit at index value of count to character

for loop to count through the indexes of the letterArray array

if character equals ignore case letterArray at index count1

add the current value of hashValue to the integer value of valueArray at index count1 and assign to hashValue

end for loop for letterArray

end for loop for nameSplit

return the value of hashValue mod by the value of hashTable2 length to the calling environment

end getHashValueForTable2 method

**public void buildHashTable2 (Presidents thePresident)**

declare a new TableDataItems object called tableData and pass thePresident to the constructor

declare hashValue variable and assign the return from getHashValueForTable2

while hashTable2 at hashValue not equal null and hashTable2 at hashValue not logically deleted

add 1 to the value of hashValue and assign to hashValue

if hashValue equals to hashTable1 length

mod hashValue by hashTable2 length (to do a wraparound) and assign to hashValue

end while loop

assign tableData to hashTable2 at hashValue index

end buildHashTable2 method

**public void displayHashTable1 ()**

display hash table 1 header to the user

display the headings to the user

for loop to count through the indexes of the hashTable1 array

display index number format

if hashTable1 at value of count is not equal null

if hashTable1 at value of count calling getDeletedData from TableDataItems is equal to -1

display an asterisk with the last name of the president to the user

else

display last name of president to the user

end if not null for count for loop

end count for loop

display a blank line to the user

end displayHashTable1 method

**public void alternativeDisplayHashTable1 ()**

display alternative display hash table 1 header to the user

display the headings to the user

for loop to count through hashTable1 for outer loop

display index number to user

if hashTable1 at value of out is not equal null

for loop to count through hashTable1 for the inner loop

if hashTable1 at inner loop value is not equal to null

if hashValue from getHashValueForTable1 from hashTable1 at inner equals to out and hashTable1 at inner calling getDeletedData method not equal to -1

display last name to the user

if hashValue from getHashValueForTable1 from hashTable1 at inner equals to out and hashTable1 at inner calling getDeletedData method equals -1

display asterisk before last name to the user

end if not null for inner for loop

end inner for loop

end if not null for outer for loop

end outer for loop

display blank line to the user

end alternativeDisplayHashTable1 method

**public void displayHashTable2 ()**

display hash table 2 header to the user

display the headings to the user

for loop to count through the indexes of the hashTable2 array

display index number format

if hashTable2 at value of count is not equal null

if hashTable2 at value of count calling getDeletedData from TableDataItems is equal to -1

display an asterisk with the last name of the president to the user

else

display last name of president to the user

end if not null for count for loop

end count for loop

display a blank line to the user

end displayHashTable2 method

**public void alternativeDisplayHashTable2 ()**

display alternative display hash table 2 header to the user

display the headings to the user

for loop to count through hashTable2 for outer loop

display index number to user

if hashTable2 at value of out is not equal null

for loop to count through hashTable2 for the inner loop

if hashTable2 at inner loop value is not equal to null

if hashValue from getHashValueForTable2 from hashTable2 at inner equals to out and hashTable2 at inner calling getDeletedData method not equal to -1

display last name to the user

if hashValue from getHashValueForTable2 from hashTable2 at inner equals to out and hashTable2 at inner calling getDeletedData method equals -1

display asterisk before last name to the user

end if not null for inner for loop

end inner for loop

end if not null for outer for loop

end outer for loop

display blank line to the user

end alternativeDisplayHashTable2 method

**public void deleteFromHashTable1 (Presidents thePresident)**

declare found variable of type String and assign the string Not Found to found

declare numberOfProbes of type int and assign 0 to numberOfProbes

declare temp of type Presidents and assign thePresident to temp

declare lastName of type String and assign temp calling getPresidentLastName to lastName

declare updateCode of type String and assign temp calling getPresidentUpdateCode to updateCode

declare hashValue and assign the value returned from getHashValueForTable1 method

while hashTable 1 at hashValue is not equal null

increment numberOfProbes by 1

if hashTable1 at hashValue calling getPresidentLastName method equals lastName and hashTable1 at hashValue calling getDeletedData method not equal -1

assign the string Found to found

hashTable1 at hashValue calling changeDeletedDataValue passing -1 to the method

break

end if

add 1 to hashValue and assign to hashValue

if hashValue equals the length of hashTable1

mod hashValue by hashTable1 length and assign to hashValue

end while loop

if found not equal to the string Found

assign -1 to numberOfProbes

assign updateCode to updateStatisticsType array

assign last name to updateStatisticsLastName array

assign numberOfProbes to updateStatisticsCountTree1 array

end deleteFromHashTable1 method

**public void updateInsertTable1 (Presidents thePresident)**

make new tableData object passing thePresident to the constructor

declare numberOfProbes of type int and assign 1 to numberOfProbes

declare hashValue and assign the return from getHashValueForTable1 method to hashValue

while hashTable1 at hashValue not equal to null and hashTable1 at hashValue not logically deleted

increment numberOfProbes by 1

add 1 to hashValue and assign to hashValue

if hashValue equals to the length of hashTable1

mod hashValue by the length of hashTable1 and assign to hashValue

end while loop

declare lastName of type String and assign the presidents last name to lastName

declare updateCode of type String and assign the presidents update code to updateCode

assign updateCode to updateStatisticsType array

assign lastName to updateStatisticsLastName array

assign numberOfProbes to updateStatisticsCountTree1 array

end updateInsertTable1 method

**public void deleteFromHashTable2 (Presidents thePresident)**

declare found variable of type String and assign the string Not Found to found

declare numberOfProbes of type int and assign 0 to numberOfProbes

declare temp of type Presidents and assign thePresident to temp

declare lastName of type String and assign temp calling getPresidentLastName to lastName

declare updateCode of type String and assign temp calling getPresidentUpdateCode to updateCode

declare hashValue and assign the value returned from getHashValueForTable2 method

while hashTable 2 at hashValue is not equal null

increment numberOfProbes by 1

if hashTable1 at hashValue calling getPresidentLastName method equals lastName and hashTable2 at hashValue calling getDeletedData method not equal -1

assign the string Found to found

hashTable2 at hashValue calling changeDeletedDataValue passing -1 to the method

break

end if

add 1 to hashValue and assign to hashValue

if hashValue equals the length of hashTable2

mod hashValue by hashTable2 length and assign to hashValue

end while loop

if found not equal to the string Found

assign -1 to numberOfProbes

assign updateCode to updateStatisticsType array

assign last name to updateStatisticsLastName array

assign numberOfProbes to updateStatisticsCountTree2 array

end deleteFromHashTable2 method

**public void updateInsertTable2 (Presidents thePresident)**

make new tableData object passing thePresident to the constructor

declare numberOfProbes of type int and assign 1 to numberOfProbes

declare hashValue and assign the return from getHashValueForTable2 method to hashValue

while hashTable2 at hashValue not equal to null and hashTable2 at hashValue not logically deleted

increment numberOfProbes by 1

add 1 to hashValue and assign to hashValue

if hashValue equals to the length of hashTable2

mod hashValue by the length of hashTable2 and assign to hashValue

end while loop

declare lastName of type String and assign the presidents last name to lastName

declare updateCode of type String and assign the presidents update code to updateCode

assign updateCode to updateStatisticsType array

assign lastName to updateStatisticsLastName array

assign numberOfProbes to updateStatisticsCountTree2 array

end updateInsertTable2 method

**public void displayStatistics ()**

for loop to count through all update statistics arrays which are the same length

display last name to the user

if updateStatisticsType at index count equals string A

display String Add to the user

if updateStatisticsType at index count equals string D

display String Delete to the user

if updateStatisticsCountTree1 at index count equals -1

display String Not Found to the user

if updateStatisticsCountTree1 at index count is greater than or equal to 0

display the integer value of updateStatisticsCountTree1 to the user

if updateStatisticsCountTree2 at index count equals -1

display String Not Found to the user

if updateStatisticsCountTree2 at index count is greater than or equal to 0

display the integer value of updateStatisticsCountTree2 to the user

end for loop

display blank line to the user

end displayStatistics method

**public int getHashValueForTable3 (String name)**

declare hashValue variable of type int and assign the value of 0 to the variable

declare nameSplit String array and assign the input string calling the split method and passing an empty string to the method

declare character variable of type String

for loop to count through the indexes of nameSplit array

assign nameSplit at index value of count to character

for loop to count through the indexes of the letterArray array

if character equals ignore case letterArray at index count1

add the current value of hashValue to the integer value of valueArray at index count1 and assign to hashValue

end for loop for letterArray

end for loop for nameSplit

return the value of hashValue mod by the value of hashTable3 length to the calling environment

end getHashValueForTable3 method

**public void buildHashTable3 (Presidents thePresident)**

make a new tableData object passing thePresident to the constructor

make a new theLink object passing tableData to the constructor

declare hashValue and assign the value returned from getHashValueForTable3 method

insert theLink in hashTable3 at the value of hashValue (inserting into the linked list)

end buildHashTable3 method

**public void displayHashTable3 ()**

display hash table 3 header to the user

display the headings to the user

for loop to count through the indexes of hashTable3

display the index to the user

if not hashTable3 at index out calling isLinkedListEmpty method

declare current of type Link and assign hashTable3 at index out calling getFirst method

while current not equal null

if current getDeletedData equals to -1

display an asterisk before the last name to the user

else

display the last name to the user

end while current not equal null loop

end if for out for loop

end if for loop

display blank line to the user

end displayHashTable3 method

**public void updateInsertHashTable3 (Presidents thePresident)**

make a new tableData object passing thePresident to the constructor

make a new theLink object passing tableData to the constructor

declare hashValue and assign the value returned from getHashValueForTable3 method

insert theLink in hashTable3 at the value of hashValue (inserting into the linked list)

end updateInsertHashTable3 method

**public void updateDeleteHashTable3 (Presidents thePresident)**

declare temp of type Presidents and assign thePresident to temp

declare lastName of type String and assign president last name to lastName

declare hashValue of type integer and assign the return from getHashValueForTable3 method to hashValue

declare current of type Link and assign hashTable3 at index hashValue calling getFirst method to current

while current not equal null

if current president last equals lastName

current calling getLinkData calling changeDeletedDataValue passing -1

assign current calling getNextLink method to current

end current not equal null while loop

end updateDeleteHashTable3 method

end HashTable class